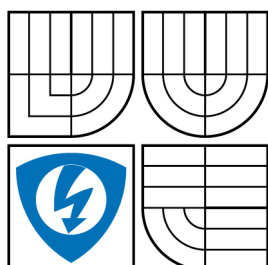


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ
ÚSTAV TELEKOMUNIKACÍ**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS**

ALGORITMY SEGMENTOVANÉ WAVELETOVÉ TRANSFORMACE OBRAZŮ

ALGORITHMS FOR SEGMENTED WAVELET IMAGE TRANSFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

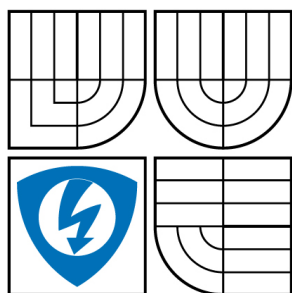
AUTOR PRÁCE
AUTHOR

PETR KOŘÍNEK

VEDOUcí PRÁCE
SUPERVISOR

Mgr. PAVEL RAJMIC Ph.D.

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a
komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor

Teleinformatika

Student: Petr Kořínek
Ročník: 3

ID: 78922
Akademický rok: 2008/2009

NÁZEV TÉMATU:

Algoritmy segmentované waveletové transformace obrazů

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte základy waveletové transformace, zejména se zaměřením na diskrétní obrazové signály. Důkladně pochopte rozklad obrazu na jeho kmitočtová pásma pomocí kombinace filtrů typu horní propust a dolní propust.

Navrhnete zobecnění jednorozměrné metody SegDTWT, která slouží k bezchybnému výpočtu waveletové transformace po segmentech, na dvojrozměrnou verzi. Navrženou metodu optimalizujte a implementujte v C++. Testujte na rozsáhlých obrazech a kvantifikujte výhody oproti klasickým segmentačním přístupům.

DOPORUČENÁ LITERATURA:

- [1] VARGIC, R: Wavelety a banky filtrů. STU, Bratislava, 2004. ISBN 80-227-2093-3
- [2] RAJMIC, P.: Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu. PhD Thesis, Brno university of Technology, Brno, 2004.

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Mgr. Pavel Rajmic, Ph.D.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Abstrakt

Tato bakalářská práce se zabývá waveletovou transformací signálu se zaměřením na obrazové signály. První část je věnována základním metodám zpracování signálu. Přes Fourierovu transformaci se dostává k základům waveletové transformace. Popisuje jednorozměrnou segmentovanou konečnou diskrétní waveletovou transformaci (SegDTWT). Pomocí této metody se odvozuje dvourozměrná segmentovaná konečná diskrétní waveletová transformace 2D signálu (SegDTWT2D).

Stěžejní částí práce je návrh a implementace této metody, která kromě dalších výhod umožňuje zpracovávat dvourozměrný signál.

Klíčová slova:

WT, waveletová transformace, DTWT, diskrétní konečná waveletová transformace, SegDTWT, segmentovaná diskrétní konečná waveletová transformace, SegDTWT2D, segmentovaná diskrétní konečná waveletová transformace 2D signálu

Abstract

The thesis deals with wavelet signal transform with the intention of image signal. First part of this work contains the basic methods for signal processing. Fourier transformation and principles of wavelet signal transform is discussed, the fundamentals of segmented discrete-time wavelet transform (SegDTWT) and the segmented discrete-time wavelet 2D signal transform (SegDTWT2D) derivation are described as well.

Main purpose of this thesis is in the segmented discrete-time wavelet 2D signal transform (SegDTWT2D) concept and its implementation which can work with two dimension signal.

Keywords:

WT, wavelet transform, DTWT, discrete-time wavelet transform, SegDTWT, segmented discrete-time wavelet transform, SegDTWT2D, segmented discrete-time wavelet 2D signal transform

Bibliografická citace mé práce:

KOŘÍNEK, P. *Algoritmy segmentované waveletové transformace obrazů*.
Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních
technologií, 2009. 34 s. Vedoucí bakalářské práce Mgr. Pavel Rajmic, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Algoritmy segmentované waveletové transformace obrazů jsem vypracoval samostatně pod vedením vedoucího bakalářské práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

podpis autora

Poděkování

Děkuji Mgr. Pavlu Rajmicovi Ph.D. za vedení bakalářské práce, odborné konzultace a cenné rady při zpracování práce.

Obsah

Úvod.....	8
1 Pojem waveletová transformace a její využití	8
2 Cíle bakalářské práce.....	8
3 Fourierova transformace.....	8
3.1 Fourierova transformace aperiodického signálu.....	9
3.2 Fourierova transformace diskrétního signálu.....	9
3.3 Krátkodobá Fourierova transformace	10
4 Waveletová transformace.....	10
4.1 Spojitá waveletová transformace.....	11
4.2 Konečná diskrétní waveletová transformace.....	11
4.3 Mallatův pyramidový algoritmus.....	12
5 Segmentovaná konečná diskrétní waveletová transformace.....	13
5.1 Metoda SegDTWT.....	14
6 Segmentovaná konečná diskrétní waveletová transformace 2D signálu - SegDTWT2D...15	
6.1 Vstupní signál a segmenty.....	15
6.2 Rozšíření segmentu.....	16
6.3 Odstranění redundantních koeficientů.....	18
6.4 Algoritmus transformace segmentu $S[R,S]$	19
6.5 Algoritmus SegDTWT2D.....	20
7 Vlastní implementace programu.....	23
7.1 Grafické uživatelské rozhraní.....	23
7.2 Konzolové rozhraní.....	25
7.3 Implementace metody SegDTWT2D.....	25
7.3.1 Schéma implementace metody SegDTWT2D.....	25
7.3.2 Úprava vstupních parametrů metody SegDTWT2D.....	28
7.3.3 Průchod maticí segmentů.....	28
7.3.4 Výstup metody SegDTWT2D do „xml“ souboru.....	29
7.3.5 Výstup metody SegDTWT2D do „txt“ souboru.....	30
Závěr.....	31
Seznam použitých zkratk, veličin a symbolů.....	32
Literatura.....	33
Seznam příloh.....	34

Úvod

1 Pojem waveletová transformace a její využití

Waveletová (vlnková) transformace, dále jen WT, se v praxi objevuje od konce 70. let minulého století. Kořeny WT však sahají až do roku 1910, kdy A. Haar zkonstruoval první wavelet [1]. A. Haar hledal alternativu k fourierovskému systému, neboť tento systém poskytuje dobrou rozlišovací schopnost ve frekvenční oblasti a horší v časové oblasti a naopak. V dnešní době WT slouží k analýze a zpracování diskrétních i spojitých signálů. Má lepší rozlišovací schopnost než klasická Fourierova transformace (FT) i její modifikace STFT (Short Time Fourier Transform).

2 Cíle bakalářské práce

Cílem práce je modifikovat jednorozměrnou metodu segmentované konečné diskrétní waveletové transformace (SegDTWT) [2] na dvourozměrnou verzi (SegDTWT2D). V úvodních kapitolách se zabývám přehledem metod pro zpracování signálu. Dále pak popisuji metodu SegDTWT, kterou pak formou návrhu a implementace v jazyku C++ rozšiřuji na dvourozměrnou metodu SegDTWT2D. Touto metodou je možné transformovat obrázky typu „bmp“ případně i jiných typů. Výstupem metody jsou pak waveletové koeficienty do dané hloubky dekompozice. Výstup se ukládá do souboru typu „xml“ a „txt“. Program má konzolové a intuitivní grafické rozhraní.

3 Fourierova transformace

Fourierova transformace (FT) slouží pro převod signálů z časové do frekvenční oblasti. Inverzní transformací získáme z frekvenční oblasti oblast

časovou. Fourierovu transformaci rozdělujeme na transformaci se spojitým a diskretním časem. Níže uvádím vztahy pro výpočet přímé a inverzní transformace.

3.1 Fourierova transformace aperiodického signálu

Fourierovu transformaci signálu se spojitým časem vypočítáme podle následujících vztahů:

Dopředná FT aperiodického signálu

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (3.1)$$

Zpětná FT aperiodického signálu

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (3.2)$$

3.2 Fourierova transformace diskretního signálu

Fourierovu transformaci diskretního signálu spočítáme pomocí následujících vztahů.

Dopředná FT diskretního signálu

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-j \frac{2\pi}{N} kn}, \quad (3.3)$$

kde N je délka posloupnosti s.

S(k) je FT diskretního signálu s(n).

Zpětná FT diskrétního signálu

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) e^{j \frac{2\pi}{N} kn}, \quad (3.4)$$

kde N je délka posloupnosti S .

$s(n)$ je zpětná transformace k $S(k)$

FT rozkládá signál na spektrální složky. Z tohoto rozkladu nejsou patrné spektrální charakteristiky v časové oblasti. Poskytuje jen informaci o frekvenčních složkách. Pokud potřebujeme získat podrobnější informace o spektru signálu v čase používá se krátkodobá FT.

3.3 Krátkodobá Fourierova transformace

Rozdíl mezi klasickou a krátkodobou FT (STFT) spočívá v tom, že STFT používá okénkovou funkci. Tato funkce rozdělí signál do více tzv. okének, což jsou stejně velké časové intervaly. Okénka se mohou překrývat, nebo nemusí být těsně vedle sebe. V okénkách se provede FT, tím dostáváme časově frekvenční vlastnosti signálu, které se znázorňují v časově-frekvenční rovině (TF rovina). Rozměry okénka jsou konstantní.

Šířka okénka je důležitý parametr STFT. Při použití úzkého okénka je dobrá rozlišovací schopnost v časové a horší ve frekvenční oblasti. U širokého okénka je dobrá rozlišovací schopnost ve frekvenční a horší v časové oblasti. Uvedené zákonitosti popisuje princip neurčitosti. Vystává otázka, jestli je možné signál popsat s velmi dobrou rozlišovací schopností ve frekvenční i časové rovině. Nejpřesnější je použití více úrovněvé analýzy, např. waveletové transformace.

4 Waveletová transformace

Waveletová transformace (WT) používá speciální funkce, tzv. wavelety (z angl. wavelet = vlnka) pro rozklad a zpětnou transformaci signálu. WT má všechny

funkce vytvořené z jedné funkce, mateřského waveletu $\psi(t)$, pomocí dvou operací, a to změna měřítka a posun v čase.

4.1 Spojitá waveletová transformace

Spojitá waveletová transformace (CWT) signálu se spojitým časem je definována vztahem [2]:

$$CWT_f(a, b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt, \quad (4.1)$$

pro $a \in \mathbb{R} \setminus \{0\}$, $b \in \mathbb{R}$.

Pokud časově-frekvenční okénko $\psi(t)$ má rozměry σ_t, σ_ω , pak u waveletu $\psi\left(\frac{t-b}{a}\right)$ se mění velikost okénka [1]:

$$\sigma_t \Rightarrow a\sigma_t, \sigma_\omega \Rightarrow \sigma_\omega / a. \quad (4.2)$$

Pro $a > 1$ se wavelet roztáhne, čímž se zmenší rozlišovací schopnost v čase, ale získáme lepší rozlišovací schopnost ve frekvenci. Pro $a < 1$ je tomu přesně naopak. Parametr b určuje možné posunutí waveletu, rozlišovací schopnost se však nemění. Ze vztahu (4.2) vyplývá, že rozlišovací schopnost v čase i frekvenci je dána pouze parametrem a . Obsah časově-frekvenčního okénka zůstává konstantní, nezáleží zde na parametrech a, b .

U CWT, na rozdíl od STFT, se mění rozměry okna a tím je SWT efektivnější při detekování signálů s vysokými frekvencemi a analýze signálů s nízkými frekvencemi.

4.2 Konečná diskrétní waveletová transformace

Pro diskrétní signály konečné délky používáme konečnou diskrétní waveletovou transformaci (DTWT). Pro výpočet používáme speciální ortogonální

matici W řádu $n \times n$ [2]. Necht' y je vstupní vektor délky n , jeho konečnou diskretní waveletovou transformací je vektor d délky n [2]:

$$d = Wy. \quad (4.3)$$

Inverzní transformaci, díky ortogonalitě matice W , lze jednoduše zapsat vztahem [2]:

$$y = W^T d. \quad (4.4)$$

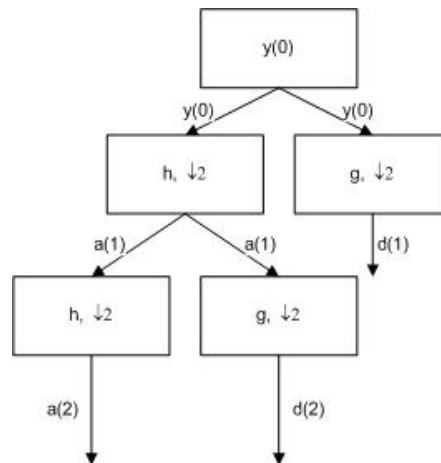
4.3 Mallatův pyramidový algoritmus

Tento algoritmus, který pochází od S. Mallata, slouží také k určení DTWT. Násobení vektoru y maticí W nahrazuje Mallatovým pyramidovým algoritmem. Matici W nahradil filtrem g , typu horní propust, a filtrem h , typu dolní propust. Vstupní signál (vektor y) projde filtrem h a filtrem g . Výstupní signály (vektory) dále decimujeme tak, že vypustíme každý druhý vzorek. Toto je jeden krok dekompozičního algoritmu. Označíme jej indexem k . Počet takových kroků označujeme jako hloubku dekompozice d . Výstup za filtrem g nazýváme detailní koeficienty d , za filtrem h aproximační waveletové koeficienty a . Pokud rozkládáme dále, takto získaný vektor za filtrem g a dále decimovaný je vstupním vektorem dalšího kroku algoritmu. Počet kroků dekompozičního algoritmu není libovolný, nazýváme jej hloubkou dekompozice, která musí splňovat vztah [1]:

$$d \leq \log_2 s, \quad (4.5)$$

kde s je délka vstupního vektoru.

Mallatův pyramidový algoritmus waveletové dekompozice je zobrazen na obr. 4.2. Detailní koeficienty většinou dále neaproximujeme, neboť potřebujeme získat nízko-frekvenční signál, odfiltrováním signálu vysokofrekvenčního.



Obr. 4.2: Mallatův pyramidový algoritmus waveletové dekompozice o hloubce $d=2$

Existuje také algoritmus waveletové rekonstrukce, jako inverzní k waveletové dekompozici. V jednom kroku rekonstrukce nejprve nadvzorkujeme vektor aproximačních popřípadě detailních koeficientů tak, že vložíme nulu za každý prvek vektoru (inverze k decimaci). Následně filtrujeme tyto vektory filtry \tilde{g} a \tilde{h} , inverzní k g a h . Vektory sečteme. Popsal jsem jeden krok rekonstrukce. Algoritmus se provádí tak dlouho, dokud se nedostaneme na úroveň původního signálu. Tento algoritmus je v praxi použitelnější než DTWT pomocí násobení maticí, díky menším výpočetním nárokům.

5 Segmentovaná konečná diskretní waveletová transformace

Již z názvu je zřejmé, že u této metody se DTWT provádí po segmentech – částech vstupního signálu. Tato metoda má své opodstatnění především při zpracování signálů v reálném čase, nebo při zpracování signálů s velkou délkou, kdy je počítačová technika omezena pamětí, nebo výpočetním výkonem. Nejprve se zmíním o metodě SegDTWT, z které vycházíme při návrhu algoritmu SegDTWT2D.

5.1 Metoda SegDTWT

Metoda provádí segmentovanou konečnou diskretní waveletovou transformace 1D signálu. Získáváme jí stejný výsledek, jako kdybychom provedli DTWT na celý vstupní signál. Proto se jednotlivé segmenty vhodným způsobem rozšiřují zleva a zprava hodnotami ze segmentu předešlého (zleva) a následujícího (zprava). Výjimku tvoří první a poslední segment, ty se rozšiřují jiným způsobem (viz níže). Důležitým parametrem pro rozšíření je délka filtru m . Následující vztahy vychází přímo z metody SegDTWT [2]. Pro návaznost koeficientů v úrovni k je třeba aby sousední koeficienty měly

$$r(k) = (2^k - 1)(m - 1) \quad (5.1)$$

společných vzorků. Pravé rozšíření segmentu N je označeno R_{\min} a levé prodloužení následujícího segmentu $L_{\max}(N+1)$. Je patrné, že:

$$r(d) = R_{\min}(N) + L_{\max}(N+1). \quad (5.2)$$

Délku segmentu včetně svého levého prodloužení označíme $l(N)$. Pro rozšíření o P_{\min} se nejprve určí hodnota rozšíření následujícího segmentu:

$$L_{\max}(N+1) = l(N) - 2^d \text{ceil}\left(\frac{l(N) - r(d)}{2^d}\right), \quad (5.3)$$

kde ceil je funkce, která zaokrouhlí na celé číslo směrem nahoru.

Pak můžeme určit hodnotu rozšíření zprava segmentu N :

$$P_{\min} = r(d) - L_{\max}(N+1) \quad (5.4)$$

Rozšíření prvního segmentu zleva se provádí o $r(d)$ vzorků. U pravého rozšíření posledního segmentu se provádí dvojité rozšíření, neboť segment může být kratší než ostatní segmenty. Nejprve se u pravého rozšíření posledního segmentu rozšíří o tolik vzorků, aby celková délka segmentu byla

$$L_{\max}(n) + s, \quad (5.5)$$

kde s je délka nerozšířeného obecného segmentu, zadaného uživatelem.

Pak se ještě poslední segment rozšíří zprava o $r(d)$ vzorků.

Po rozšíření segmentu se provádí modifikovaná WT tohoto rozšířeného. Poté se odstraní redundantní koeficienty, které vznikají rozšířením segmentu. Některé prvky segmentu se totiž transformují dvakrát.

6 Segmentovaná konečná diskretní waveletová transformace 2D signálu - SegDTWT2D

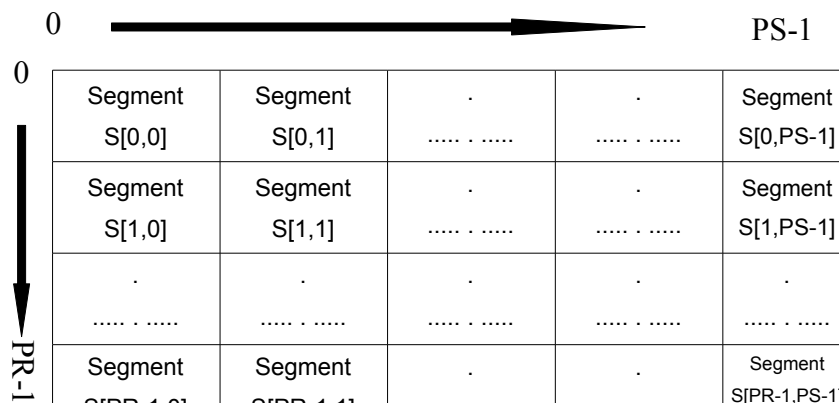
Jak již bylo uvedeno v úvodu, cílem této práce je modifikace algoritmu SegDTWT na dvourozměrnou verzi SegDTWT2D. Modifikovaná metoda navíc umožňuje zpracovávat 2D signál, což je v tomto případě obrázek. Používá jiný typ rozšíření na okrajích signálu. Jednorozměrná metoda používá rozšíření nulovými vzorky, zatímco u dvourozměrné verze se u rozšíření okrajových segmentů signálu používá tzv. zrcadlení.

Nejprve je nutno získat obrazová data ze souboru jako vstupní 2D signál. Obrázek, velikost segmentů a hloubku dekompozice volí uživatel. Následně se vstupní signál rozdělí na segmenty, které se pak vhodným způsobem rozšíří a transformují. Odstraní se redundantní koeficienty. Koeficienty transformace se uloží. Před uvedením celého algoritmu SegDTWT2D vysvětlíme jeho základní bloky.

6.1 Vstupní signál a segmenty

Obrazová data (vstupní signál) bude reprezentovat matice hodnot. Příslušnou matici rozdělíme na segmenty, čímž získáme matici segmentů $S[PR,PS]$. PS udává počet sloupců matice, PR počet řádků. Každý segment $S[R,S]$ je nyní určen indexy R a S . Hodnota R udává pozici řádku a S pozici sloupce (viz obr. 6.1). Hodnota R je

v rozsahu 0 až PR-1, hodnota S je v rozsahu 0 až PS-1. Velikost segmentu je konstantní kromě následujících výjimek. Okrajové segmenty na pravé straně matice S[R,PS-1] nebo okrajové segmenty v dolní části matice S[PR-1,S] mohou být menší (viz obr. 6.1). To přirozeně vychází z volby velikosti segmentu. Délku segmentu (uvažováno v horizontálním směru) včetně svého levého prodloužení označíme jako $l(S[R,S])$, výšku segmentu (ve vertikálním směru) včetně svého horního prodloužení jako $h(S[R,S])$.



Obr. 6.1: Matice segmentů S[R,PS]

6.2 Rozšíření segmentu

Segment rozšiřujeme horizontálním (zleva, zprava) i vertikálním směrem (nahoru, dolů). Opět platí, že sousední segmenty, v horizontálním i vertikálním směru, musí mít $r(k)$ společných vzorků (viz vztah (5.1)).

Nejprve rozšiřujeme horizontálně, pak vertikálně. Levé ($L_{\max}(S[R,S+1])$) a pravé ($P_{\min}(S[R,S])$) rozšíření je obdobné jako u SegDTWT je dáno vztahy (5.3) a (5.4). Horní rozšíření $T_{\max}(S[R-1,S])$ volíme obdobně jako $L_{\max}(N+1)$, dolní rozšíření $B_{\min}(S[R,S])$ jako P_{\min} . Vztahy pro výpočet těchto hodnot jsou uvedeny níže a jsou obdobou vztahů (5.3) [2] a (5.4) [2]. Výšku segmentu včetně svého levého prodloužení označíme $h(S)$.

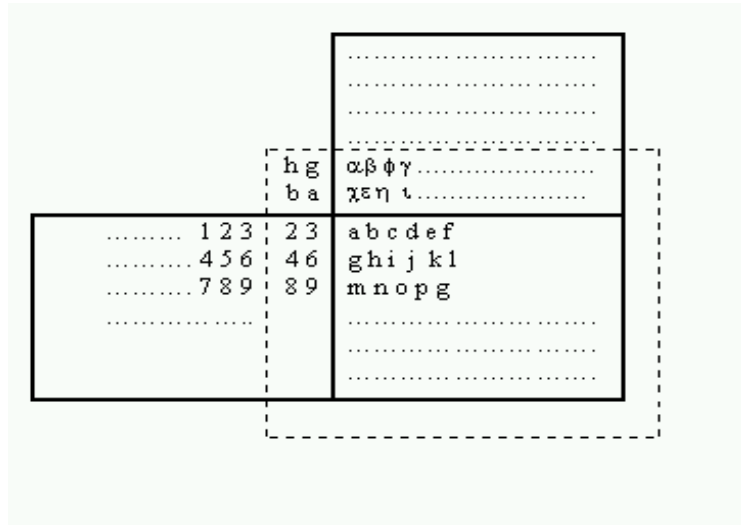
$$L_{\max}(S[R,S+1]) = l(S[R,S]) - 2^d \text{ceil}\left(\frac{l(S[R,S]) - r(d)}{2^d}\right) \quad (6.1)$$

$$P_{\min}(S[R, S]) = r(d) - L_{\max}(S[R, S+1]) \quad (6.2)$$

$$T_{\max}(S[R+1, S]) = h(S[R, S]) - 2^d \text{ceil}\left(\frac{h(S[R, S]) - r(d)}{2^d}\right) \quad (6.3)$$

$$B_{\min}(S[R, S]) = r(d) - T_{\max}(S[R+1, S]) \quad (6.4)$$

Rohy segmentu, po rozšíření zleva, zprava, shora a zdola, neobsahují žádné hodnoty. Je nutné tyto hodnoty doplnit zrcadlením podle osy kolmé na úhlopříčku segmentu. Na ob. 6.2. je naznačeno rozšíření zleva, shora o hodnoty sousedního segmentu, a rozšíření levého horního rohu segmentu. Spojitá čára určuje původní segment a čárkovaná pak rozšířený segment.



Obr.: 6.2 Rozšíření segmentu

Zvláštní případ tvoří okrajové segmenty $S[R, 0]$, $S[R, PS-1]$, $S[0, S]$, $S[PR-1, S]$. $S[R, 0]$ prodlužujeme zleva zrcadlením o $r(d)$ koeficientů, $S[R, PS-1]$ dvojitě zprava zrcadlením a doplnění nulami. Poslední segment na řádku může být kratší, mohlo by se stát, že bychom neměli dostatek koeficientů pro transformaci, vztah mezi hloubkou dekompozice a délkou segmentu popisuje vztah (4.5). Segment $S[0, S]$

rozšiřujeme podobně jako $S[R,0]$, ovšem seshora o $r(d)$ vzorků. $S[PR-1,S]$ rozšiřujeme logicky dvojité zdola zrcadlením a doplněním nulami.

Dvojitě rozšíření zprava (zdola) se provádí následovně. Nejprve se rozměry segmentu zvětší tak, aby jeho šířka (výška) byla rovna šířce (výšce) obecného segmentu, pak se ještě prodlouží zprava (zdola) o $r(d)$. Následně se do zvětšeného segmentu zrcadlí hodnoty, zbytek se doplní nulami.

6.3 Odstranění redundantních koeficientů

Redundantní koeficienty odstraňujeme kvůli prodloužení segmentu koeficienty sousedního segmentu, neboť tyto koeficienty jsou spočítány dvakrát. Odstraníme je tedy jen z jednoho segmentu, a to z pravého a dolního. Pokud jsme rozšiřovali zleva (shora), odstraníme zleva (shora) $r(d-k)$ koeficientů. U posledního segmentu na řádku matice $S[R,S-1]$ provádíme ještě dvojité odstranění redundantních koeficientů zprava, protože jsme dvojité přidali zprava. U posledního segmentu ve sloupci matice $S[R-1,S]$ provádíme dvojité odstranění redundantních koeficientů zdola.

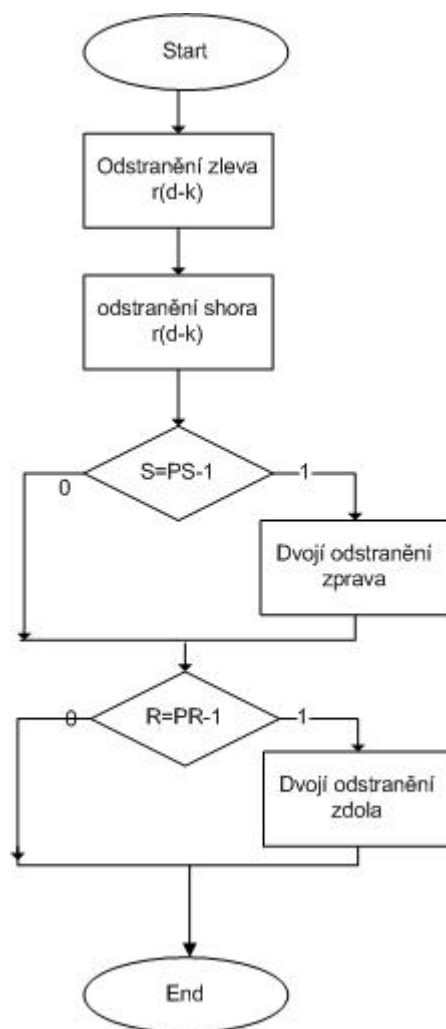
U dvojitého odstranění zprava nejprve odstraníme zprava tolik vzorků, aby v k -té úrovni zůstalo tolik vzorků, kolik uvádí vztah (6.5)[2]. Pak ještě odstraníme zprava $r(d-k)$ koeficientů. Výšku dolního segmentu označíme v' .

$$\text{floor} (2^{-k} (L_{\max} (S) + s')) \quad (6.5)$$

U dvojitého odstranění zdola odstraníme zdola tolik vzorků, aby v k -té úrovni zůstalo tolik vzorků, kolik uvádí vztah (6.6)[2]. Opět odstraníme ještě $r(d-k)$ koeficientů.

$$\text{floor} (2^{-k} (T_{\max} (S) + v')) \quad (6.6)$$

Algoritmus odstranění redundantních koeficientů je na obr. 6.3.



Obr. 6.3: Schéma algoritmu odstranění redundantních koeficientů

6.4 Algoritmus transformace segmentu $S[R,S]$

Využijeme metodu mlwd algoritmu SegDTWT, která počítá 1D transformaci. Tuto metodu musíme pozměnit, abychom ji mohli použít na 2D segment. Transformaci aplikujeme nejprve na řádky, poté z výsledného segmentu na sloupce, vždy se ale provede jen jeden krok dekompozice mlwd. Celý algoritmus transformace segmentu je uveden níže.

Algoritmus transformace segmentu $S[R,S]$:

1. krok algoritmu $k=0$;
2. jeden krok dekompozice algoritmu mlwd včetně decimace (podvzorkování dvěma) aplikovaný na řádky segmentu. Dostáváme tedy transformovanou matici S .
3. jeden krok dekompozice algoritmu mlwd včetně decimace aplikovaný na sloupce segmentu S
4. $k=k+1$
5. Pokud je k menší než hloubka dekompozice algoritmu pokračuje bodem 1., jinak algoritmus končí.

6.5 Algoritmus SegDTWT2D

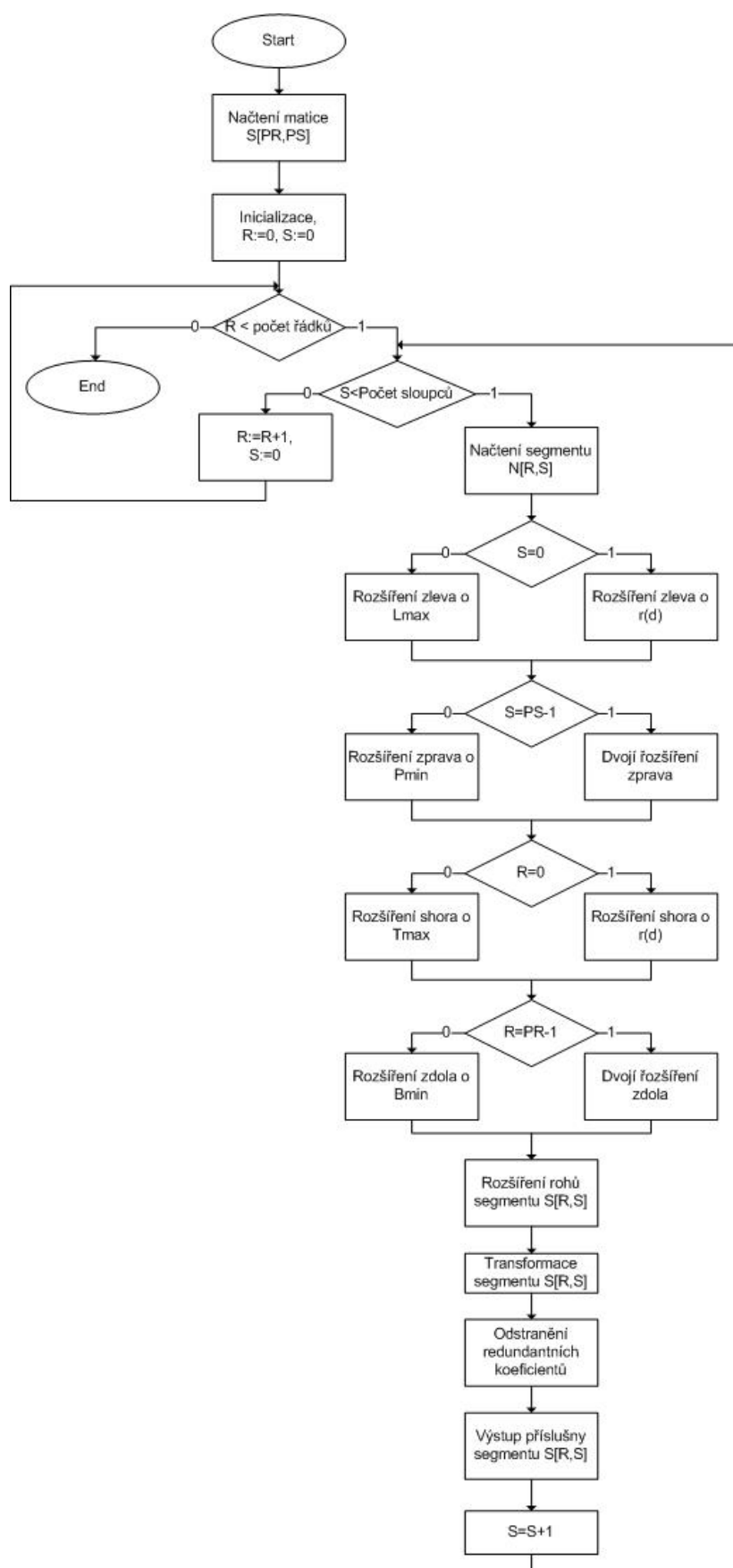
Jak již bylo napsáno výše, algoritmus provádí segmentovanou konečnou diskrétní waveletovou transformaci vstupního obrazového souboru. Popis SegDTWT2D je uveden níže. Schéma algoritmu je uvedeno na obr. 6.4.

Algoritmus SegDTWT2D:

1. Načtení matice $S[PR,PS]$
 - a) Načte se vstupní obrázek
 - b) Obrázek se rozdělí na matici segmentů $S[PR,PS]$
2. Inicializace indexů řádku a sloupce matice, přidělení nulových hodnot
3. Následující dvě podmínky řeší průchod maticí S zleva doprava a shora dolů
4. Načtení segmentu matice $S[R,S]$ (viz kapitola 6.1)
5. Následující 4 podmínky řeší rozšíření segmentu $S[R,S]$ (viz kapitola 6.2)
 - a) Pokud je segment v prvním sloupci matice, provede se rozšíření zleva zrcadlením o $r(d)$ vzorků, jinak se rozšíří zleva o L_{max} .
 - b) Pokud je segment v posledním sloupci, provedeme dvojité rozšíření zprava, jinak rozšíříme zprava o P_{min} . Vzorků.
 - c) Pokud je segment v prvním řádku matice, rozšíříme shora zrcadlením o $r(d)$ vzorků, jinak rozšíříme shora o T_{max} .
 - d) Pokud je segment v posledním řádku matice, dvojité rozšíříme zdola, jinak rozšíříme zdola o B_{min} .

6. Rozšíření rohů matice (viz kapitola 6.2).
7. Transformace segmentu $S[R,S]$ (viz kapitola 6.4).
8. Odstranění redundantních koeficientů (viz kapitola 6.3, obr. 6.3).
9. Výstup příslušný matici $S[R,S]$: uložíme koeficienty transformace.
10. Ke koeficientu sloupce přičteme 1 (pro průchod maticí S).

Algoritmus končí až projede všechny segmenty obrazu.



Obr. 6.4: Schéma algoritmu SegDTWT2D

7 Vlastní implementace programu

Pro implementaci jsem zvolil programovací jazyk C++ pro snadnou práci s paměťovým prostorem a velkou řadou vestavěných funkcí. Pro práci s maticemi jsem využil sofistikovanou knihovnu „vector“ a pro optimalizaci výpočetních úkonů knihovnu „math.h“. Pro samotnou implementaci metody SegDTWT2D a uživatelských rozhraní jsem využil vývojového nástroje „Microsoft Visual Studio 2005 Version 8.0.50727.42“. Metodu jsem implementoval pod operačním systémem „Microsoft® Windows™ XP Profesional se Service Pack 3“ běžícím v „Sun xVM VirtualBox® Version 2.1.4“.

Pro implementaci grafického rozhraní jsem využil „Microsoft .NET Framework verze 2.0“. Program v grafickém rozhraní je navržen pro operační systém „Microsoft® Windows™ XP“. Po kompilaci na „Microsoft® Windows™ 98“ a vyšším program poběží i pod těmito operačními systémy. Nezbytně nutné pro běh programu je Microsoft .NET Framework verze 2.0.

Program v konzolovém rozhraní je navržen pro operační systém „Microsoft® Windows™ XP“.

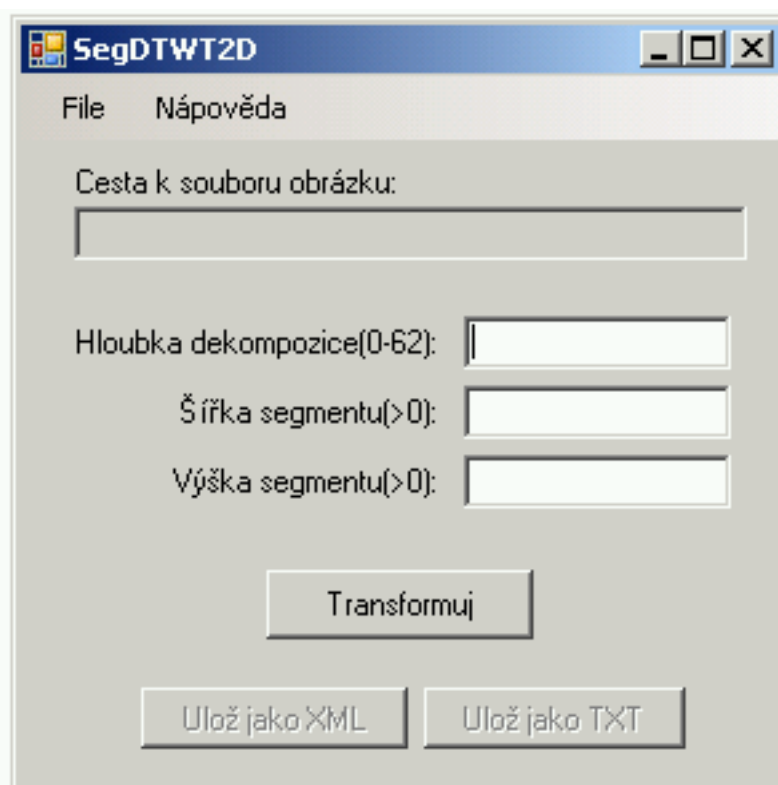
U konzolového rozhraní lze vstupní data zadávat z klávesnice, nebo přesměrovat vstupní parametry ze souboru. Což je vhodné při zpracování většího množství dat pomocí skriptu. Uživatelsky přívětivější je grafické uživatelské rozhraní, jež má podobu standardního formuláře Microsoft Windows.

7.1 Grafické uživatelské rozhraní

Pro lepší práci s programem jsem vytvořil intuitivní grafické uživatelské rozhraní. Snadno lze vybrat obrázek pro transformaci, jehož existenci ověřuje vestavěná funkce. Program zpracovává většinu běžně používaných grafických formátů. Formulář obsahuje i nápovědu k použití. Dále jsou zde 3 textová pole pro zadávání vstupních hodnot. „Hloubka dekompozice“ udává počet transformací, jež se mají aplikovat, velikost vstupu je programem omezena na čísla větší nebo rovna nule a menší nebo rovna 62. „Šířka segmentu“ určuje šířku samotného segmentu, s nímž bude program pracovat. „Výška segmentu“ následně určí výšku jednotlivého

segmentu. Obě vstupní hodnoty jsou programem přijaty jen od velikosti 1 a vyšší. Všechny zadávané vstupní hodnoty jsou akceptovány, jen pokud se jedná o číselné hodnoty.

Rozhraní obsahuje 3 tlačítka. „Transformace“ je nejdůležitější z nich, provede ověření vstupních hodnot a spustí samotný hlavní program se zadanými vstupními parametry. Po úspěšném dokončení transformace se zobrazí zpráva o provedené transformaci. Pokud jsme zadali výšku nebo šířku segmentu větší než je vstupní obrázek, program si tyto hodnoty změní na maximální přípustnou hodnotu určenou rozměry obrázku a zobrazí se zpráva o změně vstupních parametrů. Zpráva se zobrazí i pro případ, kdy zadáme větší hloubku dekompozice, než je maximální teoretická hodnota hloubky dekompozice. Rovněž nastaví tuto hodnotu na maximální možnou, která je dána metodou. Pro export do vybraných výstupních formátů slouží následující dvě tlačítka. Po výběru tlačítka „Ulož jako TXT“ nebo „Ulož jako XML“ je volána funkce, jež po výběru jména souboru vyexportuje výsledek transformace do daného souboru. Uživatelské rozhraní je zobrazeno na obr. 7.1.



Obr. 7.1: Uživatelské rozhraní

Při implementaci grafického rozhraní v „Microsoft .NET Framework verze 2.0“ se objevila chyba. Objevuje se čistě náhodně a nepředvídatelně. Je vyvolána makrem `assert`, které detekuje špatnou práci s ukazateli. Tato práce je však plně korektní a bez problému. Po ignorování této výjimky je možno dokončit výpočet bez jakéhokoli vlivu na výsledek. Program je nutné kompilovat pouze v tzv. debug módu, release mód je nestabilní. Opět je to způsobeno makrem `assert`. Debug mód programu je ovšem výrazně pomalejší a paměťově náročnější než release mód.

7.2 Konzolové rozhraní

Pro jednodušší zpracování většího množství dat jsem vytvořil konzolové rozhraní. Vstupní parametry programu, lze, kromě vstupu z klávesnice, načítat pomocí operátoru přesměrování vstupu ze souboru. Po spuštění programu jsou požadovány příslušné parametry. Povolený rozsah vstupních parametrů je totožný jako u grafického rozhraní kromě typu vstupního souboru, který je omezen na nekomprimovaný soubor typu „bmp“. Výstupní soubory „vystup.txt“ a „vystup.xml“ se ukládají do adresáře, z kterého je program spuštěn.

7.3 Implementace metody SegDTWT2D

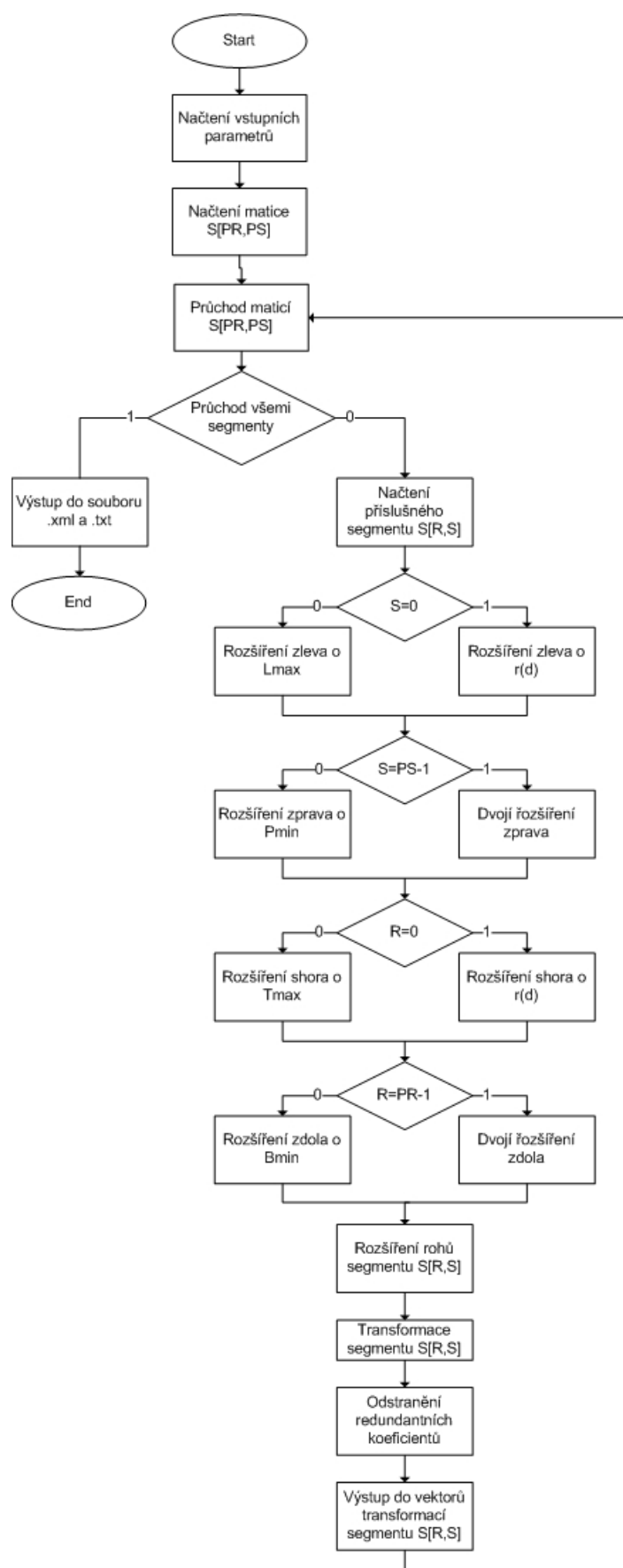
Vlastní metoda má 4 vstupní parametry zadané předané přes uživatelské rozhraní. Jsou to hloubka dekompozice, výška segmentu, šířka segmentu a absolutní cesta k vstupnímu souboru (obrázku). Korektnost formátu parametrů řeší uživatelské rozhraní. Vstupní obrazový soubor, pokud je barevný, se nejprve převede na stupně šedi (8 bit). Výstupem metody jsou výsledná transformovaná data dle příslušných vstupních hodnot. Výstup metody lze uložit ve dvou formátech: „xml“ a textovém souboru

7.3.1 Schéma implementace metody SegDTWT2D

Schéma implementace popisuje vlastní implementaci metody. Vzhledem k návrhu algoritmu SegDTWT2D došlo ke změně průchodu maticí segmentů. Schéma implementace metody je na obr. 7.2.

Implementace metody SegDTWT2D:

1. Načtení vstupních parametrů.
 - a) Načte se vstupní obrázek.
 - b) Načtou se hodnoty parametrů transformace (hloubka dekompozice, výška a šířka segmentu).
2. Obrázek se rozdělí na matici segmentů $S[PR,PS]$.
3. Průchod maticí segmentů (viz kapitola 7.2.3).
4. Pokud algoritmus zpracoval všechny segmenty, pak je výstup do souborů typu „.xml“ a „.txt“ a algoritmus končí, jinak se pokračuje bodem 5.
5. Načtení segmentu matice $S[R,S]$ (viz kapitola 6.1).
6. Následující 4 podmínky řeší rozšíření segmentu $S[R,S]$ (viz kapitola 6.2).
 - a) Pokud je segment v prvním sloupci matice, provede se rozšíření zleva zrcadlením o $r(d)$ vzorků, jinak se rozšíří zleva o L_{max} .
 - b) Pokud je segment v posledním sloupci, provedeme dvojité rozšíření zprava, jinak rozšíříme zprava o P_{min} vzorků.
 - c) Pokud je segment v prvním řádku matice, rozšíříme shora zrcadlením o $r(d)$ vzorků, jinak rozšíříme shora o T_{max} .
 - d) Pokud je segment v posledním řádku matice, dvojitě rozšíříme zdola, jinak rozšíříme zdola o B_{min} .
7. Rozšíření rohů matice (viz kapitola 6.2).
8. Transformace segmentu $S[R,S]$ (viz kapitola 6.4).
9. Odstranění redundantních koeficientů (viz kapitola 6.3, obr. 6.3).
10. Výstup do vektorů transformací segmentu.



Obr. 7.2: Schéma implementace metody SegDTWT2D

7.3.2 Úprava vstupních parametrů metody SegDTWT2D

Metoda ověřuje šířku segmentu, výšku segmentu a hloubku dekompozice. Šířka i výška nemohou přesáhnout rozměry vstupního obrázku. Pokud jsou rozměry segmentu větší než rozměry vstupního obrázku, příslušný rozměr segmentu se zmenší na odpovídající rozměr obrázku. Maximální hloubka dekompozice je určena vztahem (4.5). Navíc pro hloubku dekompozice musí ještě platit:

$$x \geq r(d), \quad (7.1)$$

kde x zastupuje výšku a šířku segmentu.

Pro rozšíření segmentu platí, že levé a horní rozšíření segmentu z předchozího segmentu (v řádku nebo ve sloupci) je maximálně $r(d)$. Rozšíření segmentu zleva a shora se musí provést vzorky z předchozího segmentu (na řádku nebo ve sloupci). Nelze rozšiřovat zleva (shora) vzorky z jiného než předchozího segmentu.

7.3.3 Průchod maticí segmentů

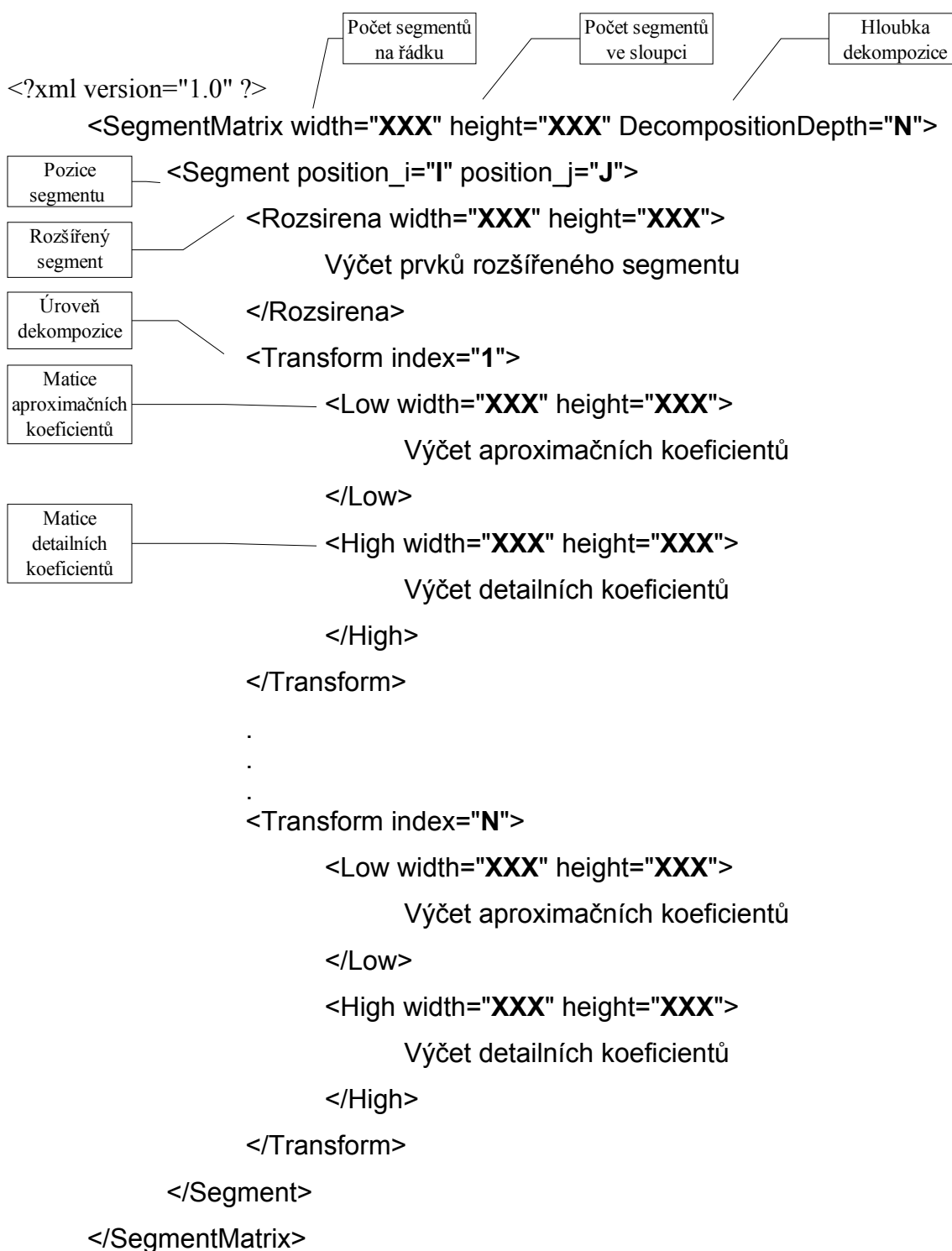
Jak již bylo naznačeno výše, ve vlastní implementaci metody se neprochází matice segmentů postupně po řádcích zleva doprava, jak je uvedeno v návrhu algoritmu. Segmenty se prochází od pravého spodního rohu matice S ($S[PR - 1, PS - 1]$). Nejdříve se zpracuje poslední řádek v matici, poté poslední sloupec. Následně se prochází segmenty v předposledním řádku a sloupci. Dále je průchod maticí segmentů zprava doleva po řádcích směrem k hornímu řádku.

Tento postup je zvolen s ohledem na optimalizaci programu. Při rozšíření segmentu rozšiřujeme vzorky z okolních segmentů. Velikost segmentů v posledním sloupci a řádku matice může být menší než u ostatních segmentů. Pokud bychom rozšiřovali segment v předposledním řádku nebo sloupci vzorky ze segmentu posledního řádku nebo sloupce, mohla by velikost segmentu, z kterého rozšiřujeme, být menší než počet potřebných vzorků. Proto bychom museli nejprve tento poslední segment na řádku nebo ve sloupci rozšířit. Pokud uvažujeme algoritmus který zpracovává segment po segmentu, nelze tedy zpracovávat dva segmenty současně.

7.3.4 Výstup metody SegDTWT2D do „xml“ souboru

XML soubor zajišťuje přehlednou orientaci ve výstupních datech a lze jej otevřít i v textovém editoru. Struktura výstupního „xml“ souboru s popisem tzv. tagů, značek a ukládaných hodnot je uvedena níže.

Struktura xml souboru:



Poznámka :

XXX – Celočíselná hodnota

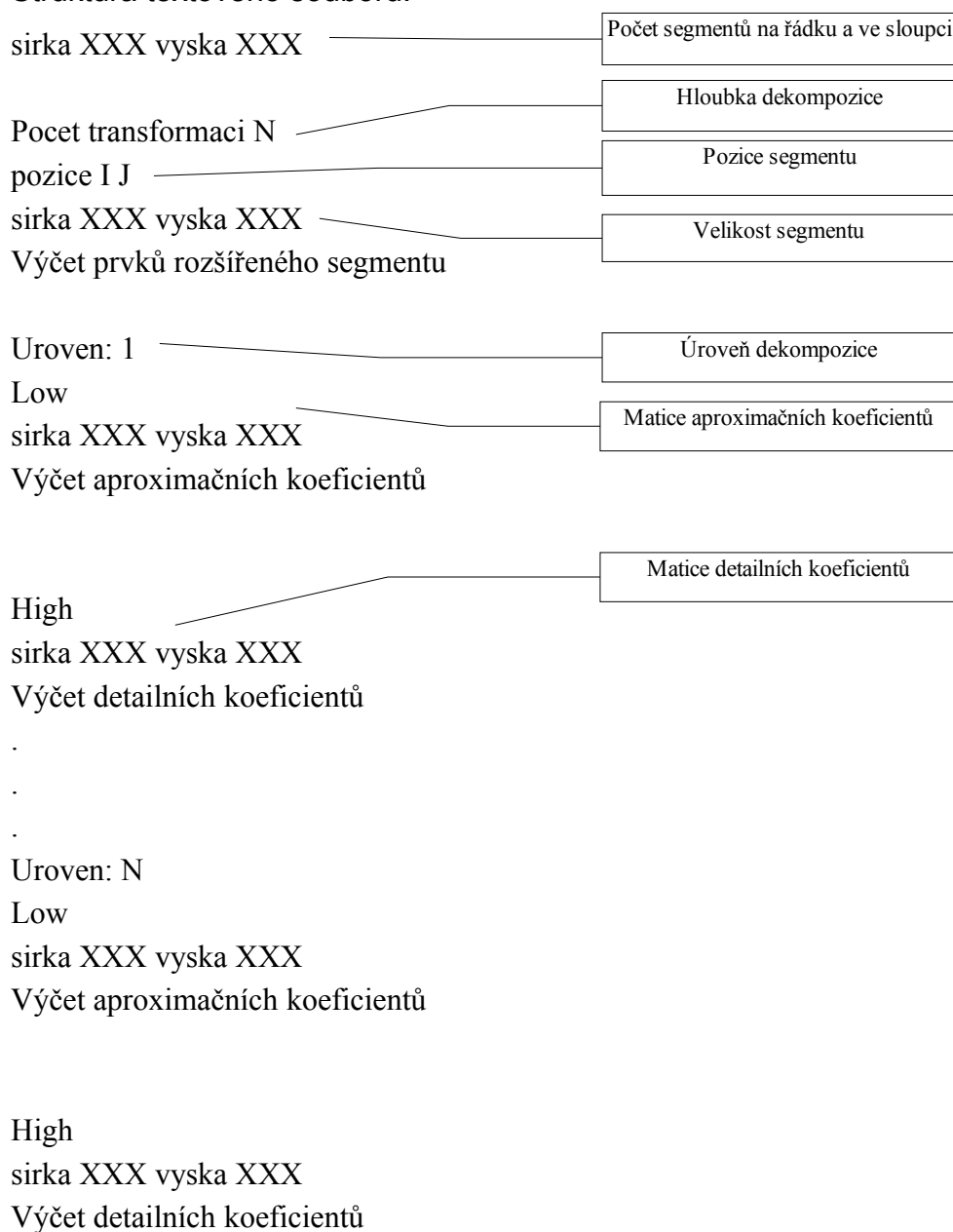
N – Hloubka dekompozice

I,J – Pozice segmentu v matici segmentů ($S[I,J]$)

7.3.5 Výstup metody SegDTWT2D do „txt“ souboru

Alternativním výstupem programu je textový soubor. Lze jej jednoduchým způsobem importovat do tabulkového procesoru „Microsoft Excel“. Oddělovačem dat je mezera a řádky. Struktura výstupního „txt“ souboru s popisem je uvedena níže.

Struktura textového souboru:



Závěr

V bakalářské práci byla prezentována dvourozměrná metoda segmentované konečné diskretní waveletové transformace obrazů (SegDTWT2D). Tato metoda byla formou návrhu a implementace algoritmu zobecněna z jednorozměrné verze.

V prvních částech bakalářské práce (po kapitole 3) bylo prezentováno zpracování spojitých i diskretních signálů. Byli uvedeny výhody i nedostatky tohoto zpracování. V kapitole 4 je pozornost věnována waveletové transformaci signálu. Je zde popsáno srovnání předchozích metod a waveletové transformace a základní charakteristiky této metody. Kapitola 5 uvádí základy algoritmu pro zpracování jednorozměrných signálů (SegDTWT). Následující kapitoly představují stěžejní část práce. Kapitola 6 pojednává o algoritmu segmentované konečné diskretní waveletové transformace dvourozměrných signálů (SegDTWT2D). Obsahuje návrh a popis tohoto algoritmu. Kapitola 7 představuje vlastní implementaci metody v jazyku C++. Ukazuje rozdíly mezi návrhem a implementací metody. Popisuje dvě uživatelská rozhraní, konzolové a grafické. Zobrazuje strukturu výstupních souborů.

V rámci bakalářské práce byl vytvořen také software pro segmentovanou diskretní konečnou waveletovou transformaci obrazů, a to ve dvou verzích. První verzí je program v grafickém, druhou v konzolovém rozhraní. Výhody a nedostatky obou verzí popisuje kapitola 7.

Velkou výhodou metody je volba velikosti segmentu. Software je poměrně dosti náročný na paměťový prostor a procesor. Rychlejší je konzolová verze, neboť verze v grafickém rozhraní je spouštěna v debug módu. Na druhou stranu program s grafickým uživatelským rozhraním je uživatelsky přívětivější a podporuje více typů obrazových souborů. Testovalo se na několika vstupních obrazových souborech programem v konzolové verzi. Odpovídající výsledky testování jsou v přílohách.

Seznam použitých zkratek, veličin a symbolů

FT	Fourierova transformace
$B_{\min}(S)$	Dolní prodloužení segmentu S
ceil (\cdot)	Zaokrouhlení na celé číslo směrem nahoru
CWT	Spojité waveletová transformace
d	Hloubka dekompozice
DTWT	Konečná diskretní waveletová transformace
floor (\cdot)	Zaokrouhlení na celé číslo směrem dolů
g	Filtr typu horní propust
h	Filtr typu dolní propust
k	Úroveň dekompozice
$L_{\max}(S)$	Levé prodloužení segmentu S
m	Délka filtru
mlwd	Jedna z metod algoritmu SegDTWT
PR	Počet řádků
PS	Počet sloupců
$R_{\min}(S)$	Pravé prodloužení segmentu S
s	Délka (šířka) segmentu
S	Matice segmentů
s'	Délka (šířka) posledního segmentu na řádku
$S[x,y]$	Prvek matice segmentů – segment
SegDTWT	Segmentovaná konečná diskretní waveletová transformace
SegDTWT2D	Segmentovaná konečná diskretní waveletová transformace 2D signálu
STFT	Krátkodobá Fourierova transformace
TF	Časově-frekvenční
$T_{\max}(S)$	Horní prodloužení segmentu S
v'	Výška posledního segmentu ve sloupci
WT	Waveletová transformace
W^T	Matice transponovaná k matici W

Literatura

- [1] VARGIC, R: *Wavelety a banky filtrov* [online]. STU, Bratislava, 2003. [cit. 2008-12-01]. Dostupné na Internetu: <http://www.ktl.elf.stuba.sk/~vargic/papers/wabf_skripta/wabf2044s.pdf>
- [2] RAJMIC, P.: *Využití waveletové transformace a matematické statistiky pro separaci signálu a šumu*. Doktorská dizertační práce, Vysoké učení technické v Brně, Brno, 2004.
- [3] WICKERHAUSER, M. V.: *Adapted wavelet analysis from theory to software* / Natick : A. K. Peters, 1994. 486 s. ISBN 1-56881-041-5

Seznam příloh

Obsah přiloženého CD

Přiložené CD obsahuje 3 adresáře:

/program v konzolovém rozhraní

obsahuje projekt Visual Studio v konzolovém rozhraní s hlavními knihovnamí algoritmu: funkce.h a funkce.cpp.

/program v grafickém rozhraní

obsahuje projekt Visual Studio v grafickém rozhraní s hlavními knihovnamí algoritmu: funkce.h a funkce.cpp.

/spustitelné soubory

obsahuje spustitelné soubory zkompilevané pro Microsoft Windows XP.

/testování

obsahuje složky s jednotlivými testy. V každé složce je vstupní obrazový soubor, soubor info.txt popisující parametry transformace a výstupní soubory transformace ve formátu „txt“ a „xml“.